PWM and PFM for visual servoing in fully decoupled approaches

Pau Muñoz-Benavent, J. Ernesto Solanes, Luis Gracia*, Josep Tornero

Abstract— **In this paper, novel visual servoing techniques based on Pulse Width Modulation (PWM) and Pulse Frequency Modulation (PFM) are presented. In order to apply previous pulse modulations, a fully decoupled position based visual servoing approach (i.e. with block-diagonal interaction matrix) is considered, controlling independently translational and rotational camera motions. These techniques, working at high frequency, could be considered to address the sensor latency problem inherent in visual servoing systems. The expected appearance of ripple due to the concentration of the control action in pulses is quantified and analyzed under simulated scenario. This high frequency ripple does not affect the system performance since it is filtered by the manipulator dynamics. On the contrary it can be seen as a dither signal to minimize the impact of friction and overcome back-lashing.**

Index Terms—**Visual servoing, robot vision systems, robot manipulators, robot control.**

I. Introduction

Visual servoing (VS) or visual servo control represents the use of feedback information extracted from a computer vision system to control the motion of a robot or any mechanical system [1]. The aim of the control scheme is to minimize the difference between the measure of a set of visual features and its desired values.

During the last three decades, many visual servoing schemes have been proposed in the literature, most of them differing in the selection of the visual features. However, all of them can be classified in two main categories, depending on the workspace in which the control is set: image-based or 2D visual servoing (IBVS) [2][3][4] and position-based or 3D visual servoing (PBVS) [5][6][26]. The velocity controller proposed in [2] relates the camera velocity with the visual features by means of the image Jacobean, namely interaction matrix.

Visual servo controllers have been typically designed to get an exponential decoupled decrease of the error under ideal conditions. Each component of the error vector has an individual convergence speed, and thus an individual convergence time, that is directly related to their respective initial errors. In the absence of constraints, the desired control strategy would be to even all the components out of the error vector to have the same convergence time, avoiding any part of the system to be overstretched. This can be accomplished introducing a diagonal weighting matrix for modifying error. Weighting matrices have been proposed in [7], [8] and [9] to combine classical PBVS and IBVS approaches, in 5D visual servoing [10] for balancing the PBVS and the IBVS schemes and in [11] to propose a robust control scheme.

Similarly to weighting matrices for modifying the error, gain matrices could be used to tune the gain associated to each of the camera velocities, but effects of motion coupling in the interaction matrix arise. That is, each factor in the weighting matrix is associated with one and only one error, but this uniqueness is lost with camera velocities if coupling in the interaction matrix exists, which makes the velocities evolution unpredictable.

Researchers in the field have proposed different visual servoing schemes trying to decouple the camera degrees of freedom (DOFs) by selecting adequate visual features. The goal is to find six features, such that each one is related only to one degree of freedom, forcing the interaction matrix to become diagonal. As stated in [12], the Grail would be to find out a diagonal interaction matrix whose elements are constant, that is as near as possible to the identity matrix, leading to a pure, direct and simple linear control problem. This goal is still pending, but steps have been done in the direction of partially decoupling camera DOFs: partitioned approach for IBVS to isolate motion related to the optic axis [13]; image moments for IBVS [14]; homography [15], [16] and epipolar geometry to decouple rotation from translation in IBVS [15]; hybrid approaches with block-triangular interaction matrix to decouple the rotational control loop from the translational one [17] or vice versa [18].

In this regard, it is noteworthy that all PBVS methods are at least partially decoupled, since the camera rotation is decoupled from camera translation, but furthermore one of the PBVS methods [19] is fully decoupled, allowing to control independently translational and rotational motions. In this case, weighting matrix and gain matrix are equivalent if they are designed as block-diagonal matrices: one block dedicated to the control of the translation motion and another one to the rotational motion.

The article proposes to transfer to the visual servoing domain the core of the well-known Pulse Width Modulation (PWM) and Pulse Frequency Modulation (PFM) techniques: obtaining an average value from signals at high frequency. PWM and PFM have been widely used for improving efficiency in power electronics devices, against noise in data transmission, and to obtain analog outputs from digital control devices. Underlying benefits can be somehow transferred to the visual servoing domain. Thus, this article is framed within the concepts of multi-rate sampling for

This work was supported in part by the Spanish Ministry of Economy and Competitiveness under Grant BES-2010-038486.

P. Muñoz-Benavent, J.E. Solanes, L. Gracia and J. Tornero are with the Instituto IDF of the Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain
(e-mails: pmunyoz,esolanes,jtornero@idf.upv.es, luigraca@isa.upv.es)
*Corresponding author: Tel. +34 963879770, luigraca@isa.upv.es

visual servoing, along the line of what the authors did in [20].

This paper is organized as follows. Next section provides background on visual servoing schemes and, specifically, the existing proposals for decoupling DOFs. Section III presents the proposal of this work: one method for errors weighting and gains tuning to control independently the decoupled camera motions and its adaptation to what we coined PWM and PFM visual servoing. The proposed approach is simulated in Section IV to show its feasibility and effectiveness, while Section V discusses several practical issues. Finally, some conclusions are given.

II. Preliminaries

A. Visual Servoing Schemes

Visual servoing schemes consist in defining a robot task by an error function to be minimized:

$$\mathbf{e} = \mathbf{s} - \mathbf{s}^* \tag{1}$$

where $\mathbf{s}$ is a vector of visual features, and $\mathbf{s}^*$ their desired values, all of them expressed in meters.

In the case of a fixed goal pose and a motionless target, s^* is constant and changes in s depend only on camera motion. In particular, the time variation $\dot{\mathrm{s}}$ of the visual features can be expressed as:

$$\dot{\mathbf{s}} = \mathbf{L}_\mathrm{s}\boldsymbol{\tau} \tag{2}$$

where $\mathbf{L}_\mathrm{s}$ is the interaction matrix related to s and $\boldsymbol{\tau} = (\mathbf{v}, \boldsymbol{\omega})$ is the camera kinematic screw. One of the most common approaches is to ensure an exponential decoupled decrease of the visual features error to their desired value s^* (i.e. $\dot{\mathbf{e}} = -\lambda\mathbf{e}$). If a free-flying camera observing a static object is considered, the corresponding control law is:

$$\boldsymbol{\tau} = -\lambda\hat{\mathbf{L}}_\mathbf{s}^+\mathbf{e} = -\lambda\hat{\mathbf{L}}_\mathbf{s}^+(\mathbf{s} - \mathbf{s}^*) \tag{3}$$

where $\hat{\boldsymbol{L}}_{\boldsymbol{s}}^+$ is an approximation of the Moore-Penrose pseudo-inverse of $\mathbf{L}_\mathrm{s}$, λ a positive scalar gain tuning the time to convergence, and $\boldsymbol{\tau}$ the camera velocities sent to the low-level robot controller.

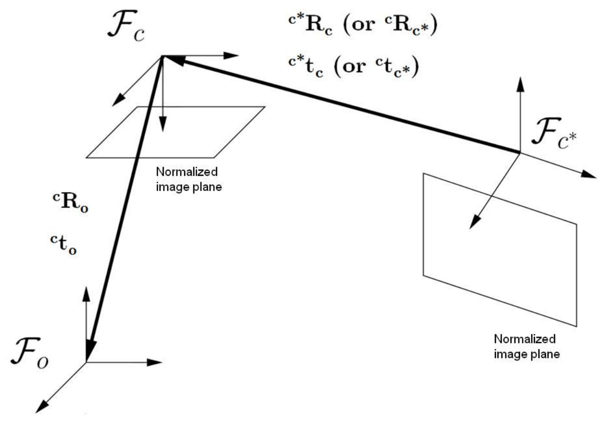


Figure 1. Coordinate frames involved in the visual servoing problem.

Figure 1 shows coordinate frames and transformations involved in the visual servoing problem. The following notation is used: the leading superscript denotes the frame with respect to which a set of coordinates (subscript) is defined. For example, the coordinate vector ${}^{c}t_{c^*}$ represents the translation coordinates of the origin of the desired camera frame expressed relative to the current camera frame.

B. Decoupling DOFs in Visual Servoing Approaches

Some different visual servoing schemes have been proposed in order to decouple DOFs, some of which are shortly described next. The problem is restricted to the classical positioning task of a free-flying camera with six DOFs with respect to a motionless target.

Approaches with block-triangular interaction matrix of the form $\boldsymbol{L_s} = \begin{pmatrix} \boldsymbol{L_{s1}} & \boldsymbol{0} \\ \boldsymbol{L_{s3}} & \boldsymbol{L_{s2}} \end{pmatrix}$ or $\boldsymbol{L_s} = \begin{pmatrix} \boldsymbol{L_{s1}} & \boldsymbol{L_{s3}} \\ \boldsymbol{0} & \boldsymbol{L_{s2}} \end{pmatrix}$, allow a decoupled behavior of the translation or rotation camera motion respectively (partially decoupled approaches), while approaches with block-diagonal interaction matrix of the form $\boldsymbol{L_s} = \begin{pmatrix} \boldsymbol{L_{s1}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{L_{s2}} \end{pmatrix}$ induce completely decoupled translational and rotational motions (fully decoupled approaches).

1) Partitioned approach to IBVS control

This approach is proposed in [13] to overcome the problem of undesired camera trajectories in Cartesian space produced in IBVS. Visual features s are defined to isolate motion related to the optical axis.

$$\dot{\mathbf{s}} = \mathbf{L}_\mathbf{s}\boldsymbol{\tau} = \mathbf{L}_\mathrm{XY}\boldsymbol{\tau}_\mathrm{XY} + \mathbf{L}_\mathrm{Z}\boldsymbol{\tau}_\mathrm{Z} = \dot{\mathbf{s}}_\mathrm{XY} + \dot{\mathbf{s}}_\mathrm{Z} \tag{4}$$

$\dot{\mathbf{s}}_\mathrm{Z} = \mathbf{L}_\mathrm{Z}\tau_\mathrm{Z}$ gives the component of $\mathbf{s}$ due to the camera motion along and rotation about the optical axis, while $\dot{\mathbf{s}}_\mathrm{XY} = \mathbf{L}_\mathrm{XY}\boldsymbol{\tau}_\mathrm{XY}$ gives the component of $\mathbf{s}$ due to velocity along and rotation about the camera X and Y axes.

Control actions associated to the optical axis are defined as:

$$\begin{cases} \mathbf{v}_\mathrm{Z} = -\lambda_{\mathrm{v_Z}} \ln\left(\frac{\sigma^*}{\sigma}\right) \\ \boldsymbol{\omega}_\mathrm{Z} = -\lambda_{\omega_\mathrm{Z}} (\alpha^* - \alpha) \end{cases} \tag{5}$$

where σ and α are two new image features defined to determine $\boldsymbol{\tau}_\mathrm{Z}$.

For X and Y axes, the resulting control action can be seen as a common IBVS control action, but with a modified error to take into account the error induced by $\boldsymbol{\tau}_\mathrm{Z}$.

$$\boldsymbol{\tau}_\mathrm{XY} = -\mathbf{L}_\mathrm{XY}^+(\lambda\mathbf{e}_\mathrm{XY} + \mathbf{L}_\mathrm{Z}\boldsymbol{\tau}_\mathrm{Z}) \tag{6}$$

where $\boldsymbol{e_{XY}}$ is the error vector associated to the common visual features in IBVS.

2) *Image moments for partially decoupled IBVS*

New visual features (image moments) are presented in [14] to decouple DOFs under IBVS. This approach leads to a block-triangular interaction matrix for all camera poses, such that the image plane is parallel to the object:

$$\boldsymbol{L}_{\boldsymbol{s}}^{||} = \begin{pmatrix} \boldsymbol{L}_{\boldsymbol{s}1}^{||} & \boldsymbol{L}_{\boldsymbol{s}3}^{||} \\ \mathbf{0} & \boldsymbol{L}_{\boldsymbol{s}2}^{||} \end{pmatrix} \tag{7}$$

Then, a generalization based on a virtual camera rotation is proposed to extend the decoupling properties for any desired camera orientation with respect to the considered object.

3) *Homography and epipolar geometry to decouple rotation from translation in IBVS*

Homography is used in [16] for planar objects to decouple rotation from translation. The proposed control law results in the following block-triangular interaction matrix:

$$\boldsymbol{\tau} = -\lambda \begin{pmatrix} \mathrm{d}^* \cdot \mathbf{M}_\mathrm{v}^{-1} & -\mathrm{d}^* \cdot \mathbf{M}_\mathrm{v}^{-1} \cdot \mathbf{M}_\omega \\ \mathbf{0} & \mathbf{I} \end{pmatrix} \begin{pmatrix} \mathbf{s} - \mathbf{s}^* \\ \mathbf{r} - \mathbf{r}^* \\ {}^{\mathrm{c}^*}\boldsymbol{\theta}\mathbf{u}_\mathrm{c} \end{pmatrix} \tag{8}$$

with $\mathrm{r} = \frac{\mathrm{d}}{\mathrm{d}^*}$ being the ratio between the current distance d and desired distance d^* to the object and ${}^{\mathrm{c}^*}\boldsymbol{\theta}\mathbf{u}_\mathrm{c}$ the rotation matrix in angle-axis representation that gives the orientation of the current camera frame relative to the desired frame.

The same approach is used in [15], but with a different control action: they generate a straight optimal trajectory by constraining the translation direction using the homography matrix. They assign camera translation to take the shortest path to the goal and camera rotation to keep the object in the field of view, and control them separately. Another algorithm proposed in [15] applies for general 3D object and uses the epipolar condition held between the goal image and the current image to generate the optimal trajectory of the robot motion to reach the goal straightforwardly.

4) *Hybrid visual servoing*

In [17] the coordinates of the central point in the image plane and the logarithm of its depth in the camera frame are used as features related to camera translation $\mathbf{s}_\mathrm{t} = (\mathrm{x}, \mathrm{y}, \log \mathrm{Z})$, $\mathbf{s}_\mathrm{t}^* = (\mathrm{x}^*, \mathrm{y}^*, \log \mathrm{Z}^*)$, $\mathbf{e}_\mathrm{t} = (\mathrm{x} - \mathrm{x}^*, \mathrm{y} - \mathrm{y}^*, \log \mathrm{Z}/\mathrm{Z}^*)$. ${}^{\mathrm{c}^*}\boldsymbol{\theta}\mathbf{u}_\mathrm{c}$ is used as feature related to rotation.

The resulting interaction matrix is block-triangular, allowing the decoupling of rotational from translational camera motions: $\dot{\mathbf{s}} = \begin{pmatrix} \mathbf{L}_\mathrm{v} & \mathbf{L}_\omega \\ \mathbf{0} & \mathbf{L}_{\theta\mathrm{u}} \end{pmatrix} \boldsymbol{\tau}$. The control law takes the form:

$$\begin{cases} \mathbf{v}_\mathrm{c} = -\mathbf{L}_\mathrm{v}^{+}(\lambda \mathbf{e}_\mathrm{t} + \mathbf{L}_\omega \boldsymbol{\omega}_\mathrm{c}) \\ \quad \boldsymbol{\omega}_\mathrm{c} = -\lambda {}^{\mathrm{c}^*}\boldsymbol{\theta}\mathbf{u}_\mathrm{c} \end{cases} \tag{9}$$

A similar approach is presented in [18], but in this case to decouple translation from rotation. Visual features vector is defined with the translational vector of the desired camera frame with respect to the current camera frame ${}^{\mathrm{c}}\mathbf{t}_{\mathrm{c}^*}$, the coordinates of the central point in the image plane (x, y), and the third component of the rotation matrix in axis-angle representation $\boldsymbol{\theta}\mathbf{u}_\mathrm{z}$. Thus, $\mathbf{s} = ({}^{\mathrm{c}}\mathbf{t}_{\mathrm{c}^*}, \mathrm{x}, \mathrm{y}, \boldsymbol{\theta}\mathbf{u}_\mathrm{z})$, $\mathbf{s}^* = (\mathbf{0}, \mathrm{x}^*, \mathrm{y}^*, \mathbf{0})$, and $\mathbf{e} = ({}^{\mathrm{c}}\mathbf{t}_{\mathrm{c}^*}, \mathrm{x} - \mathrm{x}^*, \mathrm{y} - \mathrm{y}^*, \boldsymbol{\theta}\mathbf{u}_\mathrm{z})$. The resulting interaction matrix is block-triangular:

$$\dot{\mathbf{s}} = \begin{pmatrix} {}^{\mathrm{c}^*}\mathbf{R}_\mathrm{c} & \mathbf{0} \\ \mathbf{L}_\mathrm{v}{}' & \mathbf{L}_\omega{}' \end{pmatrix} \boldsymbol{\tau} \tag{10}$$

with ${}^{c^*}\boldsymbol{R}_c$ being the rotation matrix that gives the orientation of the current camera frame relative to the desired frame.

5) *Position-based visual servoing*

Camera rotational motion control is decoupled from the translational one by definition in all common implementations of position-based visual servoing. The resulting interaction matrix is either block-triangular or block-diagonal.

We focus here on the fully decoupled PBVS approach [19], in which current visual features are set as $\mathbf{s} = ({}^{\mathrm{c}^*}\mathbf{t}_\mathrm{c}, {}^{\mathrm{c}^*}\boldsymbol{\theta}\mathbf{u}_\mathrm{c})$, desired visual features as $\mathbf{s}^* = \mathbf{0}$ and error as $\mathbf{e} = ({}^{\mathrm{c}^*}\mathbf{t}_\mathrm{c}, {}^{\mathrm{c}^*}\boldsymbol{\theta}\mathbf{u}_\mathrm{c})$. Thus, the relationship between the camera motion and the features is:

$$\dot{\mathbf{s}} = \mathbf{L}_\mathrm{s}\boldsymbol{\tau} = \begin{pmatrix} {}^{\mathrm{c}^*}\mathbf{R}_\mathrm{c} & \mathbf{0} \\ \mathbf{0} & \mathbf{L}_{\boldsymbol{\theta}\mathbf{u}} \end{pmatrix} \tau \tag{11}$$

which leads to the following control action expression:

$$\begin{cases} \mathbf{v}_\mathrm{c} = -\lambda_\mathrm{v} \, ({}^{\mathrm{c}^*}\mathbf{R}_\mathrm{c})^\mathrm{T} \; {}^{\mathrm{c}}\mathbf{t}_{\mathrm{c}^*} \\ \quad \boldsymbol{\omega}_\mathrm{c} = -\lambda_\omega \, {}^{\mathrm{c}^*}\boldsymbol{\theta}\mathbf{u}_\mathrm{c} \end{cases} \tag{12}$$

The expected behavior is described in [21]: In this case the corresponding desired feature is a null vector, and the interaction matrix is known to be block-diagonal, inducing decoupled translational and rotational motions. Moreover, each of the rotation DOFs could be controlled independently since they are directly related to one of the rotation camera motions. The corresponding camera trajectory is a straight 3D line. However, no control at all is done in the image and the visual features used for the pose estimation may be lost. This image boundary constraint is not considered in the present work.

III. Proposal

In this section, a procedure to adjust the convergence time of all the components of the error vector in visual servoing schemes is presented. The goal of this approach is to avoid overstretching on any part of the visual servoing system during the control motion task and for that, a modification of the error vector by adding a weighting matrix in the control law is proposed. Afterwards, the same procedure is applied to tune the control gains in fully decoupled visual servoing approaches. The resulting gains can be seen as two different levels of a continuous signal, what led us to explore the possibility of applying signal modulation techniques. It turned finally on what we have coined PWM and PFM visual servoing, which allows controlling independently translational and rotational camera motions with signals at high frequency.

A. Errors weighting and Gain tuning

Visual servoing schemes use normally the same scalar gain λ for all the components of the error vector $\mathbf{e}$. In that case, each component has its own convergence speed and thus, its own settling time that differs from each other according to their respective initial errors.

If the error vector has dimension m, a diagonal weighting matrix $\mathbf{W}$ with dimension m x m can be inserted in the control law in order to tune the convergence velocity of each component:

$$\mathbf{v} = -\lambda \hat{\mathbf{L}}_{\mathbf{s}}^{+} \mathbf{W} \mathbf{e} \tag{13}$$

The goal is to achieve an equal settling time for all the components of the error vector, which determines the convergence time of the entire process.

Since the control is implemented in discrete domain, the j component of the error vector at iteration i, e_j^i, is expressed as:

$$e_j^i = e_j^0 (1 - \lambda w_j T_s)^i \tag{14}$$

where e_j^0 is the j component of the initial error vector, λ the control gain, w_j the weight for the j component and T_s the sampling time.

The number of iterations needed for convergence N is,

$$N = \frac{\log\left(\frac{e_j^N}{e_j^0}\right)}{\log(1 - \lambda w_j T_s)} \tag{15}$$

where e_j^N is the desired final error threshold.

The weight w_j needed to have a given convergence time (i.e. a given number of iterations) is:

$$w_j = \frac{1 - \left(\frac{e_j^N}{e_j^0}\right)^{\frac{1}{N}}}{\lambda T_s} \tag{16}$$

Once the problem is stated, i.e. the coordinates of the initial and desired features are known, the weights w_j for each component of the error vector e_j, are computed as follows: the weight for the highest initial error is set to a maximum fixed level, the number of iterations to convergence N is calculated with equation (15), and then the remaining weights are computed with (16). Note that this is a generic procedure, valid independently of the dimension of the error vector.

Let us focus now on fully decoupled visual servoing approaches, such as the fully decoupled PBVS presented in section II.B.5). In this case translational and rotational camera motions can be controlled independently, and their respective convergence times can be adjusted with scalar coefficients, w_v and w_ω, of a translation-rotation weighting matrix $\mathbf{W}$. Due to the fact that the interaction matrix is block-diagonal, the translation-rotation weighting matrix $\mathbf{W}$ can also be expressed as a translation-rotation control gain matrix $\mathbf{K}$, positioned on the left of the interaction matrix in the control expression, equation (17). The tuning of the scalar coefficients k_v and k_ω is then equivalent to the errors weighting.

$$\mathbf{v} = -\lambda \mathbf{K} \hat{\mathbf{L}}_{\mathbf{s}}^{+} \mathbf{e} = -\lambda \hat{\mathbf{L}}_{\mathbf{s}}^{+} \mathbf{W} \mathbf{e} \tag{17}$$

$$\mathbf{K} = \begin{pmatrix} k_v \mathbf{I}_{3x3} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & k_\omega \mathbf{I}_{3x3} \end{pmatrix} \text{ and } \mathbf{W} = \begin{pmatrix} w_v \mathbf{I}_{3x3} & \mathbf{0}_{3x3} \\ \mathbf{0}_{3x3} & w_\omega \mathbf{I}_{3x3} \end{pmatrix}$$

with $k_v = w_v$ and $k_\omega = w_\omega$.

B. PWM and PFM Visual Servoing

Well-known PWM and PFM techniques can be also applied to the visual servoing control of many systems. In Figure 2 the parameters of a modulated signal are shown, being A the pulse amplitude, τ_0 the pulse width, and τ_C pulse period (or $f_c = 1/\tau_c$ pulse frequency). If the signal is supplied as input to a device or system, which has a response time much larger than τ_c, it experiences the signal as an average value:

$$V_{av} = A \frac{\tau_0}{\tau_c} \tag{18}$$

The ratio $d = \tau_0/\tau_c$ is called the duty cycle of the square wave pulses. The average value is controlled by adjusting the duty cycle. If amplitude A and pulse period τ_C are constant, the average value can be modified with the pulse width τ_0 (PWM). Otherwise, if amplitude A and pulse width τ_0 are constant, the average value can be modified with the pulse period τ_c or pulse frequency f_c (PFM).

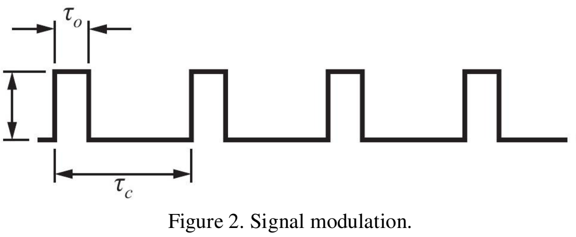


Figure 2. Signal modulation.

The existence of fully decoupled visual servoing approaches, with independent control of translational and rotational camera motions, allows us the possibility of using PWM and PFM signals for each of the independent motions. The same procedure used for errors weighting and gain tuning presented in section III.A can be applied to tune the duty cycles. Equivalence between gains and PWM-PFM duty cycles is straightforward: $k_j = d_j$. Figure 3 shows the equivalence between continuous levels of control gains and the modulated signals.

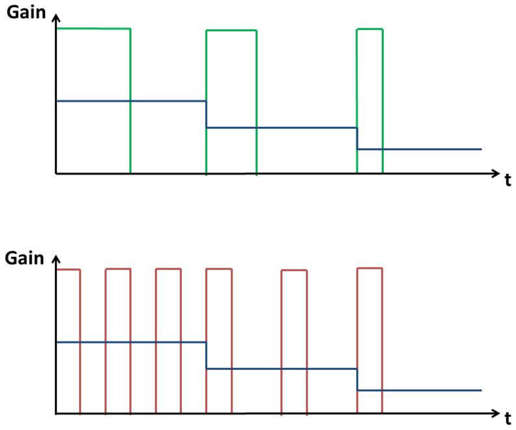


Figure 3. Gain modulation in PWM and PFM visual servoing

C. *Advantages/disadvantages of the proposed approach*

Traditional PWM and PFM benefits, which could apply to the visual servoing problem, are:

- *Discrete equivalent to analog controllers*. PWM and PFM duty cycles are adjusted to obtain an equivalent average continuous value.
- *Signal modulation in telecommunications*. PWM and PFM are forms of signal modulation: data values are encoded at one end and decoded at the other end of the transmission. In robotics, this concept can also be useful in rough industrial environments: transmitting a modulated signal instead of an analog value to avoid effects of noise.
- *Power efficiency*. PWM is used to control the amount of power delivered to a load in a more efficient way than with power delivery by resistive means. This concept of power efficiency can be transferred to the control of joint motors in robot arms. Power can be directly delivered to the power electronics stage without needing A/D conversion.

The presence of ripper in the signal due to the concentration of the control action in pulses is an expected drawback of this approach. This problem is widely analyzed and discussed in Section IV.

With the proposed control strategies, visual servoing problems are approached to the author field of knowledge, which is multi-rate control of sample-data systems [22], [23], [24].

IV. Results

In this section the proposed approach is analyzed in a simulated scenario analyzed in order to show its feasibility and effectiveness. In this sense, a comparison of the performance of different visual servoing techniques is carried out in [25] based on quantitative metrics. However, this metrics are useful to compare different approaches, but what this work actually presents are different implementations of the same approach (PBVS), so no one of that metrics is considered. The approach in [25] also categorizes work conditions that visual servo systems often experience difficulty to handle. Given that, we focus on task 1 and task 4 from [25], as we are interested in having a task involving all the 6 degrees of a free camera. Task 1 proposes to modify the initial pose through rotations in the camera Z-axis (φ_Z) between 30º and 210º. Task 4 defines axes (X_F,Y_F) lying in the feature points plane and perpendicular to the optical axis Z, and proposes initial poses resulting from rotations about X_F axis (φ_{X_F}) and Y_F axis (φ_{Y_F}) between 10º and 80º. Rotations about X_F axis imply camera translations in X and Z axes and camera rotation about Y axis, while rotations about Y_F axis imply camera translations in Y and Z axes and camera rotation about X axis.

A combination of the former two tasks is proposed: rotation of the features plane, about X_F and Y_F axes, together with a rotation around the camera optical axis Z, involving simultaneously the six DOFs of the camera. The task is evaluated under the fully decoupled PBVS approach described in section II.B.5). Euclidean norm of the initial translation and rotation errors is used to compute the desired gains and PWM-PFM duty cycles according to the procedure explained in section III.

$$\|e_0\|_2 = \left(\sqrt{{}^{c^*}\mathbf{t}_{c,X}^2 + {}^{c^*}\mathbf{t}_{c,Y}^2 + {}^{c^*}\mathbf{t}_{c,Z}^2} \quad \sqrt{{}^{c^*}\boldsymbol{\theta}\mathbf{u}_{c,X}^2 + {}^{c^*}\boldsymbol{\theta}\mathbf{u}_{c,Y}^2 + {}^{c^*}\boldsymbol{\theta}\mathbf{u}_{c,Z}^2}\right)^T \quad (19)$$

To characterize the behavior of the proposed PWM and PFM implementations, in particular to quantify the expected appearance of ripple, new error indexes are needed. The ripple can be observed in these cases: 1) in the trajectory of the visual features in the normalized image plane (2D); 2) in the time variation of the errors (3D). However, for simplicity only the first case is considered to propose the following error indexes: 1) summation of Euclidean distances, measured in the normalized image plane, between the centers of the visual features in the different approaches with respect to the number of iterations and 2) maximum value among the Euclidean distances. That is:

$$\varepsilon = \frac{\sum_{i=1}^{N} \| p_i - p_{i,GAIN} \|_2}{N} \quad (20)$$

$$\Delta = \max_i \left\{ \| p_i - p_{i,GAIN} \|_2 \right\}_1^N \quad (21)$$

being p_i the coordinates of the center of the visual features under PWM and PFM implementations, N the number of iterations, and $p_{i,GAIN}$ the coordinates of the center of the visual features in the gain tuning case, namely the non-rippled continuous case.

The experimentation is carried out in simulation under a common visual servoing scenario: positioning task of a camera with six DOFs with respect to a motionless target, object pose estimation updated with a period $\overline{\mathrm{T}}$=40 ms and the sampling period $\mathrm{T_s}$=2ms (fast enough to avoid discretization errors). The resolution of the duty cycles ($\mathrm{d_v}$, $\mathrm{d_\omega}$), and hence the available $\mathrm{d_v/d_\omega}$ and $\lambda_\mathrm{v}/\lambda_\omega$ relations are determined by $\mathrm{T_s}$:

- Under the PWM implementation, the gain pulse periods are equal to the frame period $\tau_{\mathrm{C,v}} = \tau_{\mathrm{C,\omega}} = \overline{\mathrm{T}} = 40$ ms and the gain pulse widths are $\tau_{\mathrm{0,v}} = \mathrm{n_v\,T_s}$ and $\tau_{0,\omega} = \mathrm{n_\omega\,T_s}$, in the range between $\mathrm{T_s}$ and $\overline{\mathrm{T}}$, (with $\mathrm{n_v}$ and $\mathrm{n_\omega}$ integers).

- Under the PFM implementation, the gain pulse width is equal to the sampling period $\tau_{\mathrm{0,v}} = \tau_{0,\omega} = \mathrm{T_s}$ with gain pulse periods $\tau_{\mathrm{C,v}} = \overline{\mathrm{T}}/\mathrm{n_v}$ and $\tau_{\mathrm{C},\omega} = \overline{\mathrm{T}}/\mathrm{n_\omega}$.

Rotations $\varphi_{\mathrm{X_F}}$ and $\varphi_{\mathrm{Y_F}}$ are evaluated between 10º and 60º (a combination of rotations higher than 60º in both axes leads the object to a pose non-visible from the camera) and rotation φ_Z between 30º and 180º (rotations φ_Z>180º are equivalent to rotations φ_Z-360º and rotations φ_Z=[-30,-180]º have similar behaviors to φ_Z=[30,180]º).

Table 1 and Table 2 summarize the performances for representative cases: limits in the rotation ranges and an extra case for intermediate Z axis rotation. The normalized image plane is considered infinite in order to overpass the image boundary constraint. Therefore, the effects derived from gain modulation and consequently control action modulation will be analyzed separately of other effects.

Figure 4 and Figure 5 show graphically the trajectory of the center of the visual features in the normalized image plane a), the time variation of its coordinates b), the camera translational velocities c), and the Euclidean distance of the center with respect to the gain tuning case d), in both PWM and PFM implementations, for one of the cases $(\varphi_{X_F}, \varphi_{Y_F}, \varphi_Z) = (10,60,180)$º.

By observing one of the visual servoing task $(\varphi_{\mathrm{X_F}}, \varphi_{\mathrm{Y_F}}, \varphi_\mathrm{Z}) = (10,10,30)$º, the following conclusions arise:

- Error indices increase with the initial errors in both implementations, PWM and PFM, with the same $(\varphi_{\mathrm{X_F}}, \varphi_{\mathrm{Y_F}})$ values:

$$\varepsilon(\varphi_{\mathrm{X_F}}, \varphi_{\mathrm{Y_F}}, 30) < \varepsilon(\varphi_{\mathrm{X_F}}, \varphi_{\mathrm{Y_F}}, 100) < \varepsilon(\varphi_{\mathrm{X_F}}, \varphi_{\mathrm{Y_F}}, 180)$$

$$\Delta(\varphi_{\mathrm{X_F}}, \varphi_{\mathrm{Y_F}}, 30) < \Delta(\varphi_{\mathrm{X_F}}, \varphi_{\mathrm{Y_F}}, 100) < \Delta(\varphi_{\mathrm{X_F}}, \varphi_{\mathrm{Y_F}}, 180)$$

  and also with the same φ_Z value:

$$\varepsilon(10,10,\varphi_\mathrm{Z}) < \varepsilon(10,60,\varphi_\mathrm{Z}) < \varepsilon(60,60,\varphi_\mathrm{Z})$$

$$\Delta(10,10,\varphi_\mathrm{Z}) < \Delta(10,60,\varphi_\mathrm{Z}) < \Delta(60,60,\varphi_\mathrm{Z})$$

- Error indices in PWM are greater than in the PFM implementation for the same $(\varphi_{\mathrm{X_F}}, \varphi_{\mathrm{Y_F}}, \varphi_\mathrm{Z})$ case:

$$\varepsilon_{\mathrm{PWM}}(\varphi_{\mathrm{X_F}}, \varphi_{\mathrm{Y_F}}, \varphi_\mathrm{Z}) > \varepsilon_{\mathrm{PFM}}(\varphi_{\mathrm{X_F}}, \varphi_{\mathrm{Y_F}}, \varphi_\mathrm{Z})$$

$$\Delta_{\mathrm{PWM}}(\varphi_{\mathrm{X_F}}, \varphi_{\mathrm{Y_F}}, \varphi_\mathrm{Z}) > \Delta_{\mathrm{PFM}}(\varphi_{\mathrm{X_F}}, \varphi_{\mathrm{Y_F}}, \varphi_\mathrm{Z})$$

The comparison between Δ_{PWM} and Δ_{PFM} can also be done by observing the ripple of the trajectory in the normalized image plane (Figure 4.a and Figure 5.a) and the maximum of the Euclidean distances (Figure 4.d and Figure 5.d).

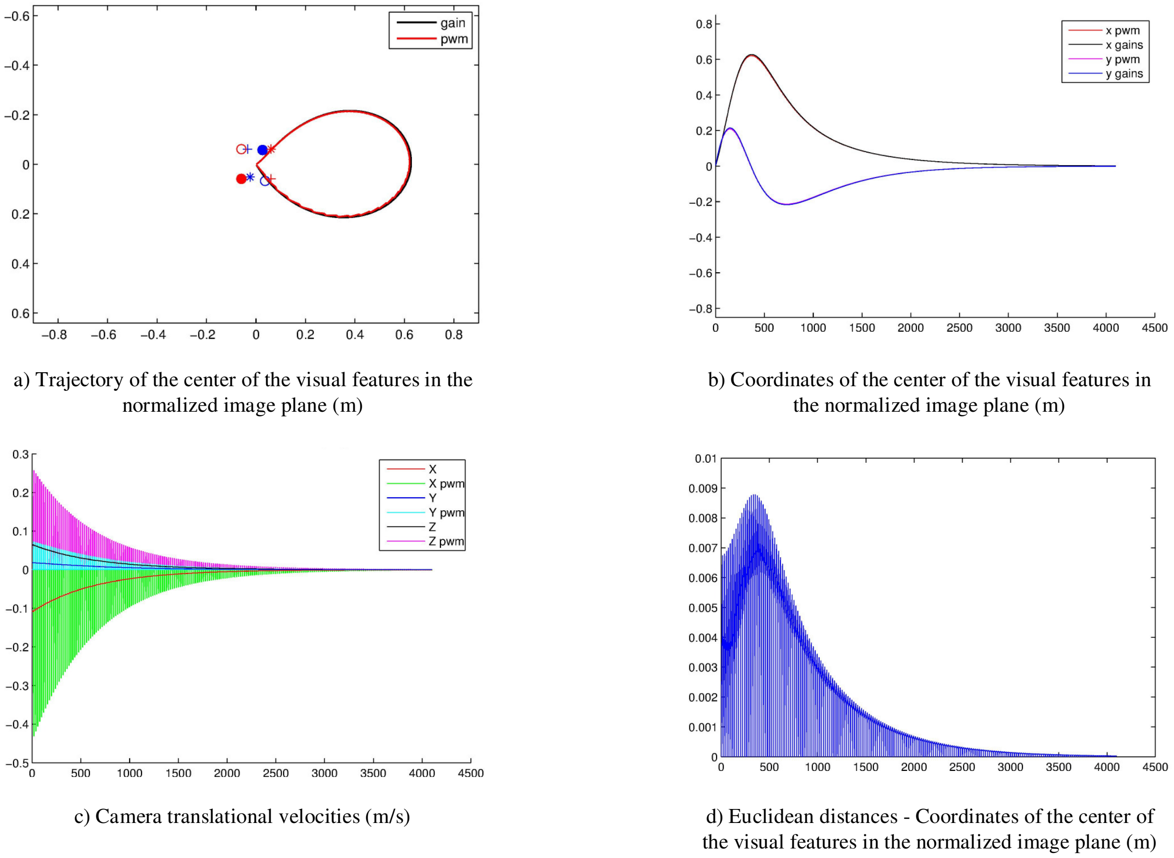
- The resulting ripper in the normalized image plane is not significant compared to the dimensions of the trajectory, as can be seen in Figure 6.

TABLE 1
PWM PERFORMANCE FOR REPRESENTATIVE CASES

$(\varphi_{X_F}, \varphi_{Y_F})$ (º)	φ_Z (º)	$(\lambda_v/\lambda_\omega)$ = (d_v/d_ω)	ε_{PWM} (x10^{-3})	Δ_{PWM}(x10^{-3})
(10,10)	30	0.75	4.1	1.8
	100	0.65	7.3	2.2
	180	0.60	12.9	4.3
(60,10) --- (10,60)	30	0.90	6.2	3.3
	100	0.80	18.6	5.9
	180	0.75	35.5	8.8
(60,60)	30	0.90	9.7	4.7
	100	0.85	29.1	7.2
	180	0.80	48.8	14.9

TABLE 2
PFM PERFORMANCE FOR REPRESENTATIVE CASES

$(\varphi_{X_F}, \varphi_{Y_F})$ (º)	φ_Z (º)	$(\lambda_v/\lambda_\omega)$ = (d_v/d_ω)	ε_{PFM} (x10^{-5})	Δ_{PFM}(x10^{-5})
(10,10)	30	0.75	5.9	1.03
	100	0.65	23.3	5.20
	180	0.60	45.4	11.33
(60,10) --- (10,60)	30	0.90	10.4	1.83
	100	0.80	62.8	13.92
	180	0.75	150	39.84
(60,60)	30	0.90	19.3	3.38
	100	0.85	75.6	17.66
	180	0.80	200	67.48



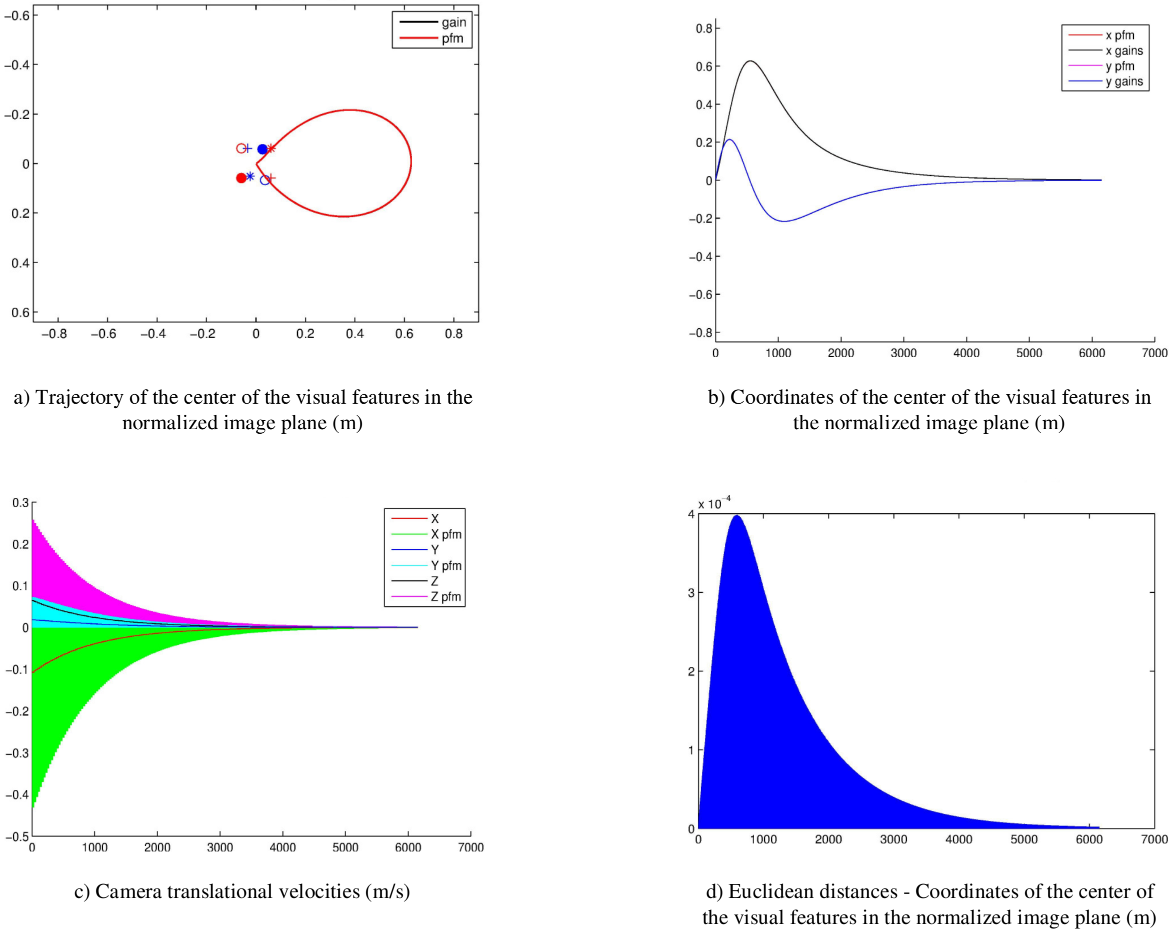
a) Trajectory of the center of the visual features in the normalized image plane (m)

b) Coordinates of the center of the visual features in the normalized image plane (m)

c) Camera translational velocities (m/s)

d) Euclidean distances - Coordinates of the center of the visual features in the normalized image plane (m)

Figure 4. Gain tuning vs. PWM implementation for $(\varphi_{X_F}, \varphi_{Y_F}, \varphi_Z) = (10,60,180)$º



a) Trajectory of the center of the visual features in the normalized image plane (m)

b) Coordinates of the center of the visual features in the normalized image plane (m)

c) Camera translational velocities (m/s)

d) Euclidean distances - Coordinates of the center of the visual features in the normalized image plane (m)

Figure 5. Gain tuning vs. PFM implementation for $(\varphi_{X_F}, \varphi_{Y_F}, \varphi_Z) = (10,60,180)$º

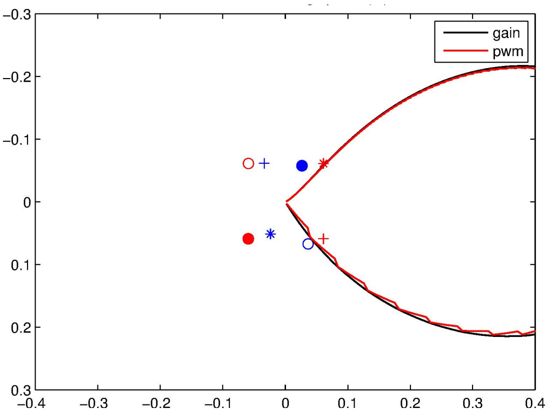


Figure 6. Zoom of the trajectory of the center of the visual features in the normalized image plane (m).

V. Practical Issues

In this section we discuss about the practical implementation and how the presence of a real robot manipulator affects the proposed method.

When a robot is taken into account, the control law is modified according to the following expression:

$$\dot{\mathbf{q}} = -\boldsymbol{\lambda}(\hat{\mathbf{L}}_{\mathbf{s}}\ {}^{c}\boldsymbol{V}_{\mathrm{e}}\ {}^{e}\boldsymbol{J}_{\mathrm{e}})^{+}\mathbf{e} \qquad (22)$$

where $\dot{\mathbf{q}}$ is the robot joints velocity vector, ${}^{c}\mathbf{V}_{\mathbf{e}}$ is the twist transformation matrix between camera and end-effector and ${}^{e}\mathbf{J}_{\mathbf{e}}$ is the robot Jacobian. The presence of these two new matrices would introduce coupling between errors and robot joints if they are not block-diagonal.

The twist transformation matrix ${}^{c}\mathbf{V}_{\mathbf{e}}$ is a fixed matrix in the case of eye-in-hand systems and is defined as follows:

$${}^{c}\mathbf{V}_{\mathbf{e}} = \begin{pmatrix} {}^{c}\mathbf{R}_{\mathbf{e}} & [\,{}^{c}\mathbf{t}_{\mathbf{e}}]_{x}\ {}^{c}\mathbf{R}_{\mathbf{e}} \\ 0_{3x3} & {}^{c}\mathbf{R}_{\mathbf{e}} \end{pmatrix}$$

Since ${}^{c}\mathbf{V}_{\mathbf{e}}$ depends on how the camera is mounted, it can be forced to be block-diagonal if $[\,{}^{c}\mathbf{t}_{\mathbf{e}}]_{x}\ {}^{c}\mathbf{R}_{\mathbf{e}} = 0$.

The robot Jacobian ${}^{e}\mathbf{J}_{\mathbf{e}}$ depends on the type of robot. Under robots with decoupled joints to end-effector transformation (such as Cartesian robots with wrist rotations), the Jacobian matrix is block-diagonal and the PWM-PFM implementation can also be applied to obtain the corresponding joint velocities $\dot{\mathbf{q}}$. However, if any coupling in the Jacobian matrix exists, the desired camera velocity would be transformed to joints velocity by means of the robot Jacobian. Discontinuities would appear in the joints velocities, but they would be filtered by the system dynamics, resulting thus in an equivalent continuous signal.

Multirate visual servoing architectures would be ideal to deal with different rates resulting from the inclusion of a robot. Information regarding the robot (i.e., the robot Jacobian) could be updated with higher rate than the visual information. Moreover, a parameterized image processing could be performed if selective information is needed, e.g., when only the translation error is needed.

In short, the proposed approach could be directly applied under Cartesian robots with wrist rotations to obtain modulated joints velocities, and multirate techniques could be applied to address the sensor latency.

VI. Conclusion

Firstly, the proposal copes with the idea of having the same convergence time for all components of the error vector by inserting an error weighting matrix in the control law without overstretching any part of the visual servoing system.

The procedure was then transferred to fully-decoupled visual servoing approaches, i.e. those with block-diagonal interaction matrix. In particular, in the fully decoupled PBVS approach, the coefficients of the gain matrix were tuned to get the same convergence time for camera translation and rotation.

Next, novel PWM and PFM visual servoing techniques were presented, consisting in modulating, in pulse width (PWM) and pulse frequency (PFM) with high-frequency signals. This opens the possibility of transferring some of the advantage of PWM and PFM to the visual servoing problem.

The expected appearance of ripple due to the concentration of the control action in pulses was analyzed under a common visual servoing scenario: a positioning task of a 6-DOFs camera with respect to a motionless object. Three main conclusions were extracted: 1) the higher the initial errors are, the higher the ripper is in both implementations; 2) the appearance of ripple is more evident in the PWM implementation; 3) the order of magnitude of the ripple is low compared with the dimensions of the signals.

This high frequency ripple does not affect to the performance since it is filtered by the dynamics of the system. Moreover, the proposed control could be used to minimize the impact of friction since it can be seen as a dither signal, a high frequency component added to the control signal to keep the system at a non-zero velocity and avoiding stick-slip friction.

With the proposed control strategies, visual servoing problems are approached to the author field of knowledge, which is multi-rate control of sample-data systems. Therefore, well-known dual-rate techniques, such as high order holds (HOH) or Lifting, could be applied.

As further work we would like to deal with the image boundary constraint by controlling independently translational and rotational camera motions, as well as to analyze the effects of non-ideal conditions in terms of pose estimation and to test the method into a real robot arm, including robot dynamics and constraints.

VII. REFERENCES

[1] L. Weiss, A. Sanderson, and C. Neuman, "Dynamic sensor-based control of robots with visual feedback," *IEEE J. Robot. Autom.*, vol. 3, no. 5, pp. 404–417, Oct. 1987.

[2] J. T. Feddema and O. R. Mitchell, "Vision-guided servoing with feature-based trajectory generation (for robots)," *IEEE Trans. Robot. Autom.*, vol. 5, no. 5, pp. 691–700, 1989.

[3] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. Robot. Autom.*, vol. 8, no. 3, pp. 313–326, Jun. 1992.

[4] K. Hashimoto, T. Kimoto, T. Ebine, and H. Kimura, "Manipulator Control with Image-Based Visual Servo," in *IEEE International Conference on Robotics and Automation*, 1991, no. April, pp. 2267–2272.

[5] W. J. Wilson, C. C. Williams Hulls, and G. S. Bell, "Relative end-effector control using Cartesian position based visual servoing," *IEEE Trans. Robot. Autom.*, vol. 12, no. 5, pp. 684–696, 1996.

[6] B. Thuilot, P. Martinet, L. Cordesses, and J. Gallice, "Position based visual servoing: keeping the object in the field of vision," in *Proceedings of the 2002 IEEE International Conference on Robtics & Automation*, 2002, no. May, pp. 1624–1629.

[7] O. Kermorgant and F. Chaumette, "Combining IBVS and PBVS to ensure the visibility constraint," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2011, pp. 2849–2854.

[8] A. H. A. Hafez and C. V. Jawahar, "Probabilistic Integration of 2D and 3D Cues for Visual Servoing," in *2006 9th International Conference on Control, Automation, Robotics and Vision*, 2006, pp. 1–6.

[9] a. Abdul Hafez and C. Jawahar, "Integration Framework for Improved Visual Servoing in Image and Cartesian Spaces," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2006, pp. 2320–2325.

[10] A. H. A. Hafez and C. V. Jawahar, "Visual Servoing by Optimization of a 2D/3D Hybrid Objective Function," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 1691–1696.

[11] A. I. Comport, E. Marchand, and F. Chaumette, "Statistically robust 2D visual servoing," *IEEE Trans. Robot.*, vol. 22, no. April, pp. 415–420, 2006.

[12] F. Chaumette and S. Hutchinson, "Visual servoing and visual tracking," in *HANDBOOK OF ROBOTICS*, B. Siciliano and O. Khatib, Eds. Springer, 2008, pp. 563–583.

[13] P. I. Corke and S. A. Hutchinson, "A new partitioned approach to image-based visual servo control," *IEEE Trans. Robot. Autom.*, vol. 17, no. 4, pp. 507–515, 2001.

[14] O. Tahri, I. I. Rennes, and C. De Beaulieu, "Image Moments : Generic Descriptors for Decoupled Image-based Visual Servo," in *IEEE International Conference on Robotics and Automation*, 2004, no. April, pp. 1185–1190.

[15] K. Deguchi, "Optimal motion control for image-based visual servoing by decoupling translation and rotation," in *Proceedings. 1998 IEEE/RSJ International Conference on Intelligent Robots and Systems. Innovations in Theory, Practice and Applications (Cat. No.98CH36190)*, 1998, vol. 2, pp. 705–711.

[16] F. Chaumette, E. Malis, and S. Boudet, "2D 1/2 visual servoing with respect to a planar object," in *IEEE International Conference on Robotics*, 1997, pp. 45–52.

[17] E. Malis, F. Chaumette, and S. Boudet, "2 1/2 D visual servoing," *IEEE Trans. Robot. Autom.*, vol. 15, no. 2, pp. 238–250, Apr. 1999.

[18] F. Chaumette and E. Malis, "2 1/2 D visual servoing: a possible solution to improve image-based and position-based visual servoings," in *Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065)*, 2000, vol. 1, pp. 630–635.

[19] P. Martinet, "Comparison of visual servoing techniques: experimental results," in *Proc. of the European Control Conference*, 1999.

[20] J. E. Solanes, L. Armesto, J. Tornero, P. Muñoz-Benavent, and V. Girbés, "Dual-rate Non-linear High Order Holds for Visual Servoing Applications.," in *Joint of the 13th Annual Conference on Towards Autonomous Robotic Systems, TAROS 2012 and the 15th Annual FIRA RoboWorld Congress*, 2012, vol. 7429, pp. 152–163.

[21] F. Chaumette and S. Hutchinson, "Visual Servo Control, Part I: Basic Approaches," *IEEE Rob Autom Mag*, vol. 13, no. 4, pp. 82–90, 2006.

[22] L. Armesto, G. Ippoliti, S. Longhi, and J. Tornero, "Probabilistic Self-Localization and Mapping - An Asynchronous Multirate Approach," *IEEE Robot. Autom. Mag.*, vol. 15, no. 2, pp. 77–88, Jun. 2008.

[23] L. Armesto, J. Tornero, and M. Vincze, “Fast Ego-motion Estimation with Multi-rate Fusion of Inertial and Vision,” *Int. J. Rob. Res.*, vol. 26, no. 6, pp. 577–589, Jun. 2007.

[24] J. Tornero and L. Armesto, “Applying Multi-Rate Modelling Techniques,” in *Advances in Systems Engineering, Signal Processing and Communication*, WSEAS, Ed. 2002, pp. 143–148.

[25] N. R. Gans, S. a. Hutchinson, and P. I. Corke, “Performance Tests for Visual Servo Control Systems, with Application to Partitioned Approaches to Visual Servo Control,” *Int. J. Rob. Res.*, vol. 22, no. 10, pp. 955–981, Oct. 2003.

[26] L. Gracia and C. Perez-Vidal, “A new control scheme for visual servoing,” *International Journal of Control, Automation and Systems*, vol. 7, no. 5, pp. 764–776, 2009.